REMARKS

Applicant has amended claim 1 to recite a machine implemented method.

The examiner rejected claims 1-21 under 35 U.S.C. 103(a) as being unpatentable over Carlson, 6,381,649 (Carlson hereafter) in view of Woo, US Pub 200210023089 (hereafter Woo).

The examiner stated:

> **As per claim 21, Carlson discloses a data monitoring and analyzing computing system that collect statistical information about network flows (abstract; col. 6, lines 38-41) comprising: a computing device that executes a computer program product stored on the computer readable medium comprising instructions to cause the computing device to (1 1, fig. 1 ; col. 5, lines 5-9; switching node 'SN' computer is the computing device that performs the data monitoring and analyzing); monitors and collects traffic flow data (i.e, accumulates packets statistics) and stores the traffic data into memory locations known as buckets (col. 7, lines 46-51 ; monitoring device has memory locations called buckets to store traffic packet data); compare the accumulated statistic values (network flow data) from the buckets to configured threshold values corresponding to the number of buckets to determine that an event is of significance (col. 3, lines 40-46; col. 7, lines 32-36 & 55-65; monitoring device compares the data units in the buckets to predetermined threshold values; out of compliance packets are 'marked' for discarding to prevent the system from excess network traffic or traffic congestion); a port to link the data collector to a central control center (20, fig. 2; col. 6, lines 38-43; input module (20, fig. 2) port links the monitoring and data collecting mechanism to the switching node (or central control device). Carlson does not explicitly disclose using a hash function to map traffic flow (packets) into the buckets and adjusting the number of buckets as the number of buckets approaches a threshold (or some pre-determined value). In an analogous art to the claimed invention, Woo discloses a packet filtering system using a hashing function to search for the packets in the index bucket table (page 5, paragraphs 0093,0096,0098; packet data is mapped using hash function in the index table (30, fig. 2); adjusting the number of bucket filters as the packet data reaches a pre-specified (threshold) value (page 1, paragraph 0023; page 4, paragraphs 0080,0081 ; as the number of packets reaches a threshold value, the number of filter buckets can change dynamically). Hence, it would have been obvious to one of ordinary skill in the art to modify and combine the teachings of Carlson and Woo to use a hash function for quick sorting or lookup and adjusting the number of buckets (or filters) to accommodate changing traffic conditions as desired by the user as disclosed by Woo on [page 5, paragraph 00911.**
>
> **Claims 1 and 14 recite similar limitations to claim 21; therefore, they are rejected using similar rationale as claim 21.**

Claim 1 is distinct over the combination of references since the references neither describe nor suggest *** producing statistics corresponding to a parameter of traffic flow to trace the source of an attack, mapping the traffic flow into a plurality of buckets by applying a hash function "f(h)" to the parameter *** to output an integer corresponding to one of the buckets, ***

comparing the number of buckets to a threshold and determining whether the number of buckets should be divided into more buckets or combined into fewer buckets based on comparing ***.

The examiner contends that Carlson discloses to "compare the accumulated statistic values (network flow data) from the buckets to configured threshold values corresponding to the number of buckets to determine that an event is of significance (col. 3, lines 40-46; col. 7, lines 32-36 & 55-65.

Applicant disagrees. Carlson at Col. 3, lines 40-46 discloses that:

> **In one embodiment, each data packet is associated with a discard eligibility value, which identifies a priority for discarding the data packet. In general, packets assigned higher discard eligibility values are more likely to be discarded if it is determined that discard is necessary for reasons such as excess data traffic or congestion. Therefore, in the present invention, data packets can be marked according to whether they cause a threshold to be exceeded by altering the discard eligibility value for the packet. That is, if a packet causes a threshold to be exceeded, the discard eligibility can be increased such that the priority for discard of the packet is increased.**

In this cited passage, Carlson teaches to discard data packets, not to configure threshold values corresponding to the number of buckets, as contended by the examiner. Rather, Carlson is directed to the problem of tracking data rates for monitoring class of service on a network. That is a distinct and non-related problem to producing statistics corresponding to a parameter of traffic flow to trace the source of an attack, as recited in claim 1.

While Carlson talks about different systems with large or small numbers of buckets, Carlson neither describes nor suggests to adjust the number of buckets in any one system. Rather, it appears that Carlson merely teaches a number of buckets based on the number of monitored links and classes of service.

Similarly, at Col. 7, lines 32-36 & 55-65 Carlson discloses that:

> **In the embodiment shown in FIG. 3, three stages of processing are used to compare a number of units of data, e.g., bytes or groups of bytes, that have been received over a particular link at a particular class of service to three predetermined threshold values. The three processing stages allow for four levels of classification of the degree to which the threshold is exceeded. Consequently, the system allows for four possible settings of the DE value associated with the packet being examined.** (Col. 7, lines 32-36)

> **In one embodiment of the invention, the stage 1 processor 50 performs a first comparison step using the bucket value stored in the first memory 56 and its corresponding threshold. If the threshold is exceeded, the DE value is incremented for the packet and the processing proceeds to the second stage. The second stage processor 52 can then perform a second comparison using the corresponding bucket value stored in memory 58 and its associated threshold. If the threshold is exceeded, the DE value can again be increased, and processing can be proceed to the third stage processor 54. If the second stage threshold is not exceeded, the present DE value at stage two can be stored back with the data packet by the processor 30 in the packet processing circuitry, and the packet with the updated DE value can be passed out of the policing circuitry 26. The third stage processor 54 can be used to perform the comparison to again increment the DE value for the packet, if needed. At any stage, if a threshold is not exceeded, the present DE value is stored back with the packet by the processor 30, and the packet is transmitted out of the policing circuitry 26 with the updated DE value.** (Col. 7, lines 55-65)

Carlson here deals with a multi stage processing arrangement to set DE values "A typical header includes a discard eligibility (DE) field and a packet length (PL) field. The DE field includes a DE value which sets the priority for discard of the packet." (Carlson col. 6 line 66 to col. 7 line 1). The DE value which is set by the process disclosed in the cited passages does not configure the number of buckets. While Carlson does teach that "corresponding memory location or bucket is adjusted or "leaked" by subtracting the present counter value from the present bucket contents," (see Abstract) Carlson has no teachings that would suggest to adjust the number of buckets, as in claim 1.

Woo does not cure the deficiencies in Carlson. Woo like Carlson, and unlike Claim 1 is simply directed to packet classification, not to producing statistics corresponding to a parameter of traffic flow to trace the source of an attack. Woo is even less relevant that Carlson, since Woo does not even suggest buckets to track statistics corresponding to traffic flow. Woo instead deals with "filter buckets." The "filter buckets" however are derived from Woo's packet classification procedure. Woo attempts to eliminate as many filters as possible filters for classification of a packet by examining specific bit positions. Woo does not eliminate all but one filter, but instead keeps a set of remaining filters that is less than some pre-specified maximum. It is this set of filters, having a maximum size, that Woo terms a "filter bucket."

The examiner uses Woo to teach:

> **a packet filtering system using a hashing function to search for the packets in the index bucket table (page 5, paragraphs 0093,0096,0098; packet data is mapped**

using hash function in the index table (30, fig. 2); adjusting the number of bucket filters as the packet data reaches a pre-specified (threshold) value (page 1, paragraph 0023; page 4, paragraphs 0080,0081 ; as the number of packets reaches a threshold value, the number of filter buckets can change dynamically).

Woo discloses the use of a hash function to search a data structure, in a search procedure where a packet is first directed to a "specific sub-tree by indexing (or hashing) via the jump table 30 using the initial prefixes of certain selected dimensions." Woo does not suggest "mapping the traffic flow into a plurality of buckets by applying a hash function "f(h)" to the parameter," as in claim 1 and thus, does not teach any of the features of Claim 1 namely *** producing statistics corresponding to a parameter of traffic flow to trace the source of an attack, mapping the traffic flow into a plurality of buckets by applying a hash function "f(h)" to the parameter *** to output an integer corresponding to one of the buckets, *** comparing the number of buckets to a threshold and determining whether the number of buckets should be divided into more buckets or combined into fewer buckets based on comparing ***.

Accordingly, assuming that it is suggested to combine the teachings of Woo and Carlson, the combination of Carlson and Woo neither describes nor suggests claim 1

Claims 2-13, which depend directly or indirectly from claim 1, are allowable at least for the reasons discussed in claim 1. Further, the claims add distinct limitations. For example, claim 3 recites that as the number of buckets changes, the buckets have values derived from the buckets prior to the change. The examiner contends that: "As per claim 3, Carlson discloses as the number of buckets changes, the buckets have values derived from the buckets prior to the change (col. 7, lines 49-54; system stores data in plurality of buckets; system maintains values of each bucket)."

Carlson discloses at (col. 7, lines 49-54) that:

Each memory has allocated a memory location or group of locations to each link and class of service being policed. These locations or groups of locations, i.e., "buckets," maintain an updatable value which can be updated upon receipt of a packet of data. Each memory 56, 58, 60 also stores the predetermined threshold for each link and class of service to which the corresponding updatable value is compared upon receipt of a new data packet.

All that Carlson teaches is that buckets maintain updatable values not that as the number of buckets change the buckets the buckets have values derived from the buckets prior to the change.

Claim 4, which recites that the hash function adapts to map to the new number of buckets, as the new number of buckets changes is also not described, since neither applying a hash function "f(h)" to the parameter *** to output an integer corresponding to one of the buckets nor changing the number of buckets is suggested by the references.

Claim 5 distinguishes since neither Carlson nor Woo suggest comparing the value accumulated in the bucket to a threshold that depends on the number of buckets. While Carlson does disclose comparing, Carlson discloses in the Abstract

> **When a data packet is received at a particular link and class of service, the corresponding memory location or bucket is adjusted or "leaked" by subtracting the present counter value from the present bucket contents. That difference is then added to the number of units of data, i.e., bytes or groups of bytes of data, contained in the incoming packet. That sum is then compared with a predetermined threshold determined by the allowable data rate associated with the link and class of service. If the threshold is exceeded, then the incoming data packet is marked accordingly.**

Carlson does not suggest comparing the value accumulated in the bucket to a threshold that depends on the number of buckets. Rather, Carlson discloses comparing to a threshold that is based on an allowable data rate associated with the link and class of service.

Claim 8 recites that the hash function changes periodically in a randomly secret manner so that packets are reassigned to different buckets. Nothing in Woo suggest this feature because Woo uses a hash to access the search tree, but randomly changing the hash would not serve any purpose in the system taught by Woo.

One of the features of Applicant's invention is to provide a variable number of buckets that dynamically adjust the amount of traffic and number of flows monitored, so that the monitoring device is not vulnerable to a denial of service attack against its own resources, as expressed in claim 9. This feature is wholly unrelated to the purposes and teachings of the cited references.

Claim 10 is distinguished, since the variable number of buckets efficiently identifies the source or sources of attack by breaking down traffic into different buckets and examining statistics accumulated for a parameter and a corresponding threshold in each bucket. Neither Carlson nor Woo teach the features of the base claim, nor teach to use the recited mechanism in detecting an attack. The references are devoid of collected data in buckets that could be used to identify a source of an attack.

Claims 11-13 contain additional distinct limitations.

Claim 14 includes instructions to map traffic flow into a plurality of buckets by applying a hash function "f(h)" to a parameter of the traffic flow to output an integer corresponding to one of the buckets, accumulate statistics from the packets, and compare the accumulated statistic values from the buckets to configured threshold values corresponding to the number of buckets to determine that an event is of significance and adjust the number of buckets as the number of buckets approaches a second threshold.

Claim 21 recites instructions to map traffic flow into a plurality of buckets by applying a hash function "f(h)" to the parameter of the traffic flow to output an integer corresponding to one of the buckets; accumulate statistics from the packets; and compare the accumulated statistic values from the buckets to configured threshold values corresponding to the number of buckets to determine that an event is of significance; and adjust the number of buckets as the number of buckets approaches a second threshold.

Claims 14 and 21 are distinct over Carlson taken separately or in combination with Woo, generally for reasons discussed in claim 1. Claim 21, for example, is distinct since the references fail to at least teach instructions to "adjust the number of buckets as the number of buckets approaches a second threshold."

Applicant has deleted the limitation of "a port to link the data collector to a central control center" from claim 21 and made that the subject of new claim 77, since this feature is not necessary to distinguish over the prior art cited.

Claims 15-20 and newly added claims 50-62 are allowable at least for the reasons discussed in their respective base claims and analogous dependent claims 2-13.

Newly added claims 63-76 are likewise allowable over the references, since the references neither describe nor suggest at least *** varying the number of buckets according to the amount of traffic and number of flows according to down traffic flow into different buckets and examining statistics accumulated for a parameter and a corresponding threshold in the bucket as in claim 63. Claim 70 is allowable for analogous reasons.

Dependent claims 64-69 add additional distinct features to claim 63. Claims 71-76 add additional distinct features to claim 70.

The art cited but not applied by the examiner is seen as neither describing nor suggesting applicant's invention.

Applicant has enclosed an Information Disclosure Statement with appropriate fee. Applicant contends that the claims are allowable over the art in the IDS and the art of record.

Enclosed is a **$60** check for the Petition for Extension of Time fee. No fees are believed due for excess claim fees, since Applicant had previously paid fees for the canceled claims. If a fee is due, please charge that fee and apply any other charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: 7/21/05

Denis G. Maloney
Reg. No. 29,670

Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110
Telephone: (617) 542-5070
Facsimile: (617) 542-8906